



JustMeds Milestone 4

Gruppo #4

[Edoardo Castagnini](#) 226613

[Carlo Zamuner](#) 227145

[Gianluigi Vazzoler](#) 227445

| | |
|---|-----------|
| 1. Sezione Introduttiva..... | 3 |
| 1.1. Project Idea..... | 3 |
| 1.2. Link repository Github..... | 3 |
| 1.3. Link ad Apiary..... | 3 |
| 1.4. Link Deployment..... | 4 |
| 2. Sezione generale..... | 4 |
| 2.1. Strategia di Branching..... | 4 |
| 2.2. Product backlog..... | 5 |
| 2.3. Definizione di “Done”..... | 8 |
| 3. Sezione Sprint #2..... | 8 |
| 3.1. Goal..... | 8 |
| 3.2. Sprint Planning..... | 9 |
| 3.2.1. Sprint Backlog..... | 9 |
| 3.2.2. Burndown Chart..... | 14 |
| 3.3. Test cases..... | 14 |
| 3.4. Sprint review..... | 22 |
| 3.5. Product backlog refinement..... | 23 |
| 3.6. Sprint retrospective..... | 24 |
| 4. Sezione Finale..... | 26 |
| 4.1. Diagramma del deploy finale con tutti i componenti (interni ed esterni)..... | 26 |
| 4.2. Stack tecnologico utilizzato..... | 27 |
| 4.3. Aggiornamento Diagramma Entità Relazione..... | 28 |
| 4.4. Conclusioni..... | 28 |

1. Sezione Introduttiva

1.1. Project Idea

L'idea è quella di un servizio web che permetta a chiunque di scegliere ed ordinare un farmaco o un prodotto farmaceutico disponibile presso le diverse farmacie della città di Trento in modo da riceverlo a casa propria in breve tempo. Da un lato, dunque, il servizio garantisce la consegna a domicilio, minimizzando gli spostamenti per coloro in difficoltà, ma dall'altro offre la possibilità, per esempio ai giovani e agli studenti, di trarre compenso svolgendo il ruolo di rider, andando quindi a ritirare il prodotto in farmacia e portandolo alla destinazione finale.

1.2. Link repository Github

Repository: [JustMeds](#)

1.3. Link ad Apiary

Apiary è un potente strumento per la documentazione delle API, in quanto fornisce un modo semplice ed efficace per descrivere e documentare le API, consentendo agli sviluppatori di comprendere facilmente come utilizzarle correttamente. Tale documentazione è fondamentale per migliorare la comunicazione nel team di sviluppo e facilitare l'integrazione con altre applicazioni.

Le API del nostro progetto sono documentate utilizzando Apiary e la documentazione completa è reperibile al seguente link: [Apiary](#). Inoltre, il file OpenAPI/Swagger (swagger.yaml) è disponibile nella repository di GitHub, per comodità al seguente link: [swagger.yaml](#)

1.4. Link Deployment

[Render deployment](#)

Account di test:

- Ricevente:
 - mail: provacliente@gmail.com
 - Password: Prova1234
- Rider:
 - mail: carlo.zamuner@gmail.com
 - Password: riderTrento

- Admin:
 - mail: admin@gmail.com
 - Password: admin
- Farmacia:
 - farmacitn@gmail.com
 - Password: FarmaciaTN

2. Sezione generale

2.1. Strategia di Branching

Nel secondo sprint, abbiamo implementato una strategia di branching più avanzata, al contrario del primo sprint, nota come Feature Branch Workflow, per migliorare l'efficienza e la gestione del nostro codice. Questa strategia prevede la creazione di branch separati per ogni nuova funzionalità, nel nostro caso in particolare, per ogni tipologia di utente. Ogni membro del team, infatti, ha lavorato su un branch dedicato, in modo che ognuno potesse sviluppare le feature e le API specifiche di quel tipo di account in isolamento, mantenendo il codice organizzato e facilitando la collaborazione all'interno del team. Grazie a tale suddivisione abbiamo evitato conflitti di codice e abbiamo mantenuto una struttura organizzata e chiara all'interno del repository.

2.2. Product backlog

In seguito allo sprint #1 il nostro product backlog è stato aggiornato con ulteriori funzionalità e specifiche che abbiamo ritenuto opportuno aggiungere al fine di migliorare il servizio web e l'esperienza utente.

Di seguito viene riportata la tabella aggiornata, consultabile anche al foglio *"product backlog after sprint #1"* del seguente [link](#)

| ID | Name | User story | Importance | How to Demo |
|----|---------|---|------------|--|
| 1 | Sign up | Come utente voglio registrarmi al servizio, come cliente o come rider, così da poter accedere alle rispettive funzionalità del servizio | 90 | Dal form di registrazione del servizio, dopo aver inserito le info richieste e premuto il pulsante di registrazione, vengo rediretto ad una pagina di account creato correttamente e, successivamente, alla pagina per effettuare il login |

| | | | | |
|---|--|---|----|--|
| 2 | Login | Come utente voglio effettuare il login al servizio per poter accedere al mio account e usufruire delle funzionalità che richiedono un utente autenticato | 90 | Dal form di autenticazione del servizio, dopo aver inserito email e password dell'account e premuto il pulsante per accedere, vengo rediretto al catalogo dei prodotti |
| 3 | Logout | Come utente loggato voglio poter effettuare il logout in qualunque momento | 70 | Dal menu a scomparsa in alto, dopo aver cliccato il tasto "logout" si viene disconnessi dal proprio account e reindirizzati alla pagina di login/registrazione |
| 4 | Form assistenza | Come utente, voglio poter accedere a un modulo di assistenza per segnalare problemi o richiedere informazioni di supporto | 40 | Dimostrare come un utente può compilare un modulo di assistenza direttamente dall'applicazione o dal sito web e inviarlo al team di supporto, che poi risponderà via email o tramite notifiche interne. |
| 5 | Visualizzazione prodotti (anche senza login) | Come utente voglio poter visualizzare la lista di farmaci disponibili che posso ordinare, senza dover necessariamente avere un account ed effettuare il login | 60 | Dal menu principale dopo aver cliccato l'apposito tasto nel menu a scomparsa, è possibile visualizzare la lista dei prodotti; per poter aggiungere i farmaci al carrello è necessario effettuare il login |
| 6 | Aggiunta prodotti al carrello | Come utente cliente voglio poter aggiungere prodotti al carrello | 80 | Dalla pagina dei prodotti è possibile aggiungerne al carrello cliccando l'apposito tasto; dopo averlo premuto si riceve una conferma della corretta aggiunta al carrello |
| 7 | Visualizzazione carrello e modifica quantità | Come utente cliente voglio poter visualizzare il carrello e i prodotti al suo interno e modificare la quantità di ciascuno di essi | 70 | Aperto il menu a tendina, dopo aver cliccato il pulsante del carrello, apparirà un popup per visualizzare e modificare la quantità di ciascun prodotto nel carrello; dopo aver cliccato sul pulsante "procedi al checkout" presente nel popup, si verrà reindirizzati alla pagina del carrello in cui sarà ancora possibile rimuovere o modificare la quantità dei prodotti e visualizzare il totale dei costi |
| 8 | Scelta farmacia | Come utente cliente voglio poter scegliere la farmacia desiderata da una lista di farmacie che supportano il servizio | 50 | Dalla pagina di checkout posso selezionare la farmacia desiderata da una lista preimpostata di farmacie |
| 9 | Accettazione finale ordine | Come utente cliente voglio poter accettare o annullare il mio ordine una volta ricevuta la conferma della farmacia e il relativo prezzo finale dell'ordine | 80 | Dalla pagina di monitoraggio ordine, una volta che l'ordine sarà arrivato alla fase di "accettato dalla farmacia", visualizzerò il prezzo finale da pagare e sarà possibile scegliere se cliccare "conferma e paga" o "annulla ordine" |

| | | | | |
|----|---|---|----|--|
| 10 | Monitoraggio ordine | Come utente cliente voglio poter monitorare il mio ordine in tutte le sue fasi | 70 | Dalla pagina del proprio account è possibile cliccare su "i miei ordini" per visualizzare e monitorare le fasi di un ordine in progresso |
| 11 | Visualizzazione e modifiche dati account | Come utente cliente voglio poter visualizzare i dati del mio account e modificare i parametri come l'indirizzo e la password | 70 | Dalla pagina del proprio account è possibile cliccare su "impostazioni" per visualizzare i propri dati personali e modificare i campi di indirizzo e password tramite gli appositi pulsanti |
| 12 | Visualizzazione storico ordini | Come utente cliente voglio poter visualizzare lo storico ordini e gli ordini attivi con i loro stati | 80 | Dalla pagina del proprio account è possibile cliccare su "i miei ordini" per visualizzare tutti gli ordini attivi e passati, suddivisi secondo i loro stati |
| 13 | Visualizzazione richieste assistenza | Come utente cliente voglio poter visualizzare le richieste di assistenza che ho mandato, ed una loro eventuale risposta | 40 | Dalla pagina del proprio account è possibile cliccare su "Richieste assistenza" per visualizzare tutte le richieste mandate ed eventuali risposte |
| 14 | Visualizzazione consegne disponibili e accettazione | Come utente rider voglio poter visualizzare la lista di ordini in attesa di ritiro con relativi indirizzi su mappe in modo da poter accettare la consegna preferita e successivamente confermare la messa a termine dell'incarico | 70 | Dalla pagina principale dell'account rider è possibile visualizzare tutti gli ordini in attesa di ritiro. Cliccandone uno è possibile accettarlo, oltre che vedere i dettagli specifici tra cui il codice segreto, da mostrare in farmacia per il ritiro. Inoltre, solo una volta accettato, sarà possibile confermare l'avvenuta consegna |
| 15 | Visualizzazione guadagni | Come utente rider voglio poter visualizzare le statistiche dei guadagni | 70 | Dalla pagina del proprio account è possibile cliccare su "I miei guadagni" per visualizzare i guadagni |
| 16 | Privilegi utente admin | Come utente admin voglio poter accedere ad azioni privilegiate come la manutenzione del sistema, visualizzazione statistiche e modifica delle impostazioni globali | 60 | Dalla pagina principale dell'account admin è possibile cliccare su diversi tasti per modificare impostazioni del servizio, visualizzare statistiche di utilizzo, feedbacks e richieste di assistenza |
| 17 | Check out: invio ordine alla farmacia | Come utente, voglio inviare l'ordine alla farmacia per confermare il prezzo. | 60 | Dimostrare l'invio dell'ordine alla farmacia e la conferma tramite pop-up. |
| 18 | Controlli lista ordini candidati | Come farmacia, voglio vedere la lista degli ordini candidati per confermare disponibilità e prezzo. | 65 | Dimostrare la visualizzazione della lista degli ordini candidati nella sezione farmacia. |
| 19 | Conferma prezzo ordine candidato | Come farmacia, voglio confermare il prezzo aggiornato finale per un ordine candidato. | 65 | Dimostrare l'inserimento del prezzo aggiornato e la conferma tramite pop-up verde. |
| 20 | Cambio stato ordine | Come farmacia, voglio aggiornare lo stato dell'ordine nelle varie fasi di selezione. | 70 | Dimostrare l'aggiornamento dello stato dell'ordine nelle varie fasi di selezione. |

| | | | | |
|----|---|---|-----|--|
| 21 | Controllo ordini pagati in attesa del Rider | Come farmacia, voglio visualizzare gli ordini pagati in attesa del passaggio del Rider. | 90 | Dimostrare la visualizzazione degli ordini pagati e in attesa di consegna dal Rider. |
| 22 | Controllo secret code | Come farmacia, voglio verificare il codice segreto per confermare la consegna al Rider. | 80 | Dimostrare l'inserimento del codice fornito dal Rider e la conferma della consegna tramite pop-up. |
| 23 | Storico ordini farmacia | Come farmacia, voglio visualizzare lo storico degli ordini consegnati e in corso. | 40 | Dimostrare la visualizzazione dello storico ordini con gli ordini consegnati e in corso. |
| 24 | Visualizzazione pool ordini Rider | Come Rider, voglio visualizzare la lista degli ordini che posso accettare. | 60 | Dimostrare la visualizzazione della lista degli ordini che il Rider può accettare. |
| 25 | Accettazione ordine da parte del Rider | Come Rider, voglio accettare un ordine e visualizzare i dettagli per il ritiro. | 60 | Dimostrare l'accettazione dell'ordine e la visualizzazione dei dettagli tramite il codice segreto. |
| 26 | Cancellazione ordine da parte del Rider | Come Rider, voglio cancellare un ordine accettato per errore. | 60 | Dimostrare la cancellazione dell'ordine da parte del Rider e il ritorno dell'ordine nella pool. |
| 27 | Consegna avvenuta | Come Rider, voglio confermare l'avvenuta consegna dell'ordine. | 70 | Dimostrare la conferma della consegna tramite l'apposito pulsante nell'applicazione. |
| 28 | Middlewares sicurezza | Come utente, voglio essere reindirizzato alla pagina di login se cerco di accedere a una pagina non autorizzata. | 100 | Dimostrare il reindirizzamento alla pagina di login quando un utente non autorizzato cerca di accedere a risorse protette. |
| 29 | Accesso con Google - utente esistente | Come utente esistente, voglio autenticarmi tramite Google e accedere alla pagina corrispondente al mio ruolo. | 100 | Come utente esistente, voglio autenticarmi tramite Google e accedere alla pagina corrispondente al mio ruolo. |
| 30 | Autorizzazione - accesso a risorsa protetta | Come utente, voglio essere reindirizzato alla pagina di login se cerco di accedere a una risorsa protetta senza autenticazione. | 100 | Dimostrare il reindirizzamento alla pagina di login quando un utente non autenticato cerca di accedere a risorse protette. |

2.3. Definizione di “Done”

Tramite la definizione di “Done” si stabiliscono i criteri per i quali si può considerare come terminata un’attività o compito, tali criteri comprendono elementi come la realizzazione del codice, il suo testing, la verifica e la redazione della documentazione. In tal senso, si migliora la trasparenza, la tracciabilità e la comunicazione all’interno del team.

Parlando di team, un task viene da noi considerato completato secondo i seguenti criteri:

- il codice deve essere testato per garantirne l’accessibilità

- deve essere stata eseguita una verifica e un test del codice da parte di un altro membro del team
- il codice deve essere adeguatamente commentato
- la documentazione relativa deve essere stata scritta e revisionata
- il codice deve essere stato caricato e il ramo di sviluppo deve essere stato integrato con l'ultima versione nel sistema di controllo versione (VCS)

3. Sezione Sprint #2

3.1. Goal

In questo secondo sprint, il più importante e vicino alla consegna, ci concentreremo sulla realizzazione completa e finale del servizio web, dalla fase di ordine dell'utente fino al pagamento e alla consegna, sia dal punto di vista del cliente, che da quello del rider e della farmacia. Ecco i punti chiave su cui lavoreremo:

1. **Fase di Ordine dell'Utente:** Implementare un sistema che consenta agli utenti di effettuare ordini e procedere con il pagamento una volta accettato dalla farmacia.
2. **Sezione Cliente:** Implementare una pagina in cui l'account cliente possa visualizzare lo storico degli ordini, i suoi dati e le sue richieste di assistenza
3. **Sezione Farmacia:** Integrare funzionalità che permettano alle farmacie di visualizzare gli ordini in arrivo, rifiutarli o accettarli confermando la disponibilità e il prezzo dei farmaci. Inoltre la farmacia dovrà confermare la consegna dell'ordine al rider verificando l'apposito codice segreto
4. **Pool di Ordini per i Rider:** Consentire ai rider di consultare la pool di ordini, selezionarne uno e procedere con il ritiro in farmacia e la consegna al cliente.
5. **Consegna al Rider:** Implementare l'uso di un codice segreto per ufficializzare il passaggio del farmaco al rider.
6. **API per il Login tramite Google:** Integrare le API per la gestione del login tramite Google per semplificare l'autenticazione degli utenti.
7. **Mappe con OpenStreetMap:** Integrare le mappe di OpenStreetMap per migliorare la logistica delle consegne.
8. **Integrazione degli Open Data:** Utilizzare gli open data nazionali del Ministero della Salute e gli open data trentini per la lista delle farmacie e dei farmaci.

3.2. Sprint Planning

3.2.1. Sprint Backlog

Al termine di questo secondo sprint siamo riusciti a concludere a pieno tutte le task che ci eravamo prefissati successivamente allo sprint #1. Di seguito viene riportata la tabella relativa allo sprint backlog #2

| | Sprint Backlog (Sprint Planning) | | | | |
|-----------|--|--|----------------|-----------|----------|
| | Name | User story | | Volunteer | Estimate |
| Sprint #2 | Accettazione finale ordine | Come utente cliente voglio poter accettare o annullare il mio ordine una volta ricevuta la conferma della farmacia e il relativo prezzo finale dell'ordine | Sviluppo UI | Edoardo | 6 |
| | | | Design API | Edoardo | 6 |
| | | | Sviluppo API | Edoardo | 7 |
| | | | Testing | Gianlugi | 4 |
| | | | Documentazione | Carlo | 2 |
| | | | Deploy | Edoardo | 2 |
| | Monitoraggio ordine | Come utente cliente voglio poter monitorare il mio ordine in tutte le sue fasi | Sviluppo UI | Edoardo | 6 |
| | | | Design API | Edoardo | 6 |
| | | | Sviluppo API | Edoardo | 7 |
| | | | Testing | Carlo | 4 |
| | | | Documentazione | Gianlugi | 3 |
| | | | Deploy | Edoardo | 2 |
| | Visualizzazione e modifiche dati account | Come utente cliente voglio poter visualizzare i dati del mio account e modificare parametri come l'indirizzo | Sviluppo UI | Edoardo | 7 |
| | | | Design API | Edoardo | 8 |
| | | | Sviluppo API | Edoardo | 8 |
| | | | Testing | Edoardo | 5 |
| | | | Documentazione | Gianlugi | 4 |
| | | | Deploy | Carlo | 2 |
| | Visualizzazione storico ordini | Come utente cliente voglio poter visualizzare lo storico ordini e gli ordini attivi con i loro stati | Sviluppo UI | Edoardo | 6 |
| | | | Design API | Edoardo | 7 |
| | | | Sviluppo API | Edoardo | 7 |
| | | | Testing | Carlo | 4 |
| | | | Documentazione | Gianlugi | 3 |
| | | | Deploy | Edoardo | 2 |

| | | | | | |
|--|---|---|----------------|-----------|---|
| | Visualizzazione richieste assistenza | Come utente cliente voglio poter visualizzare le richieste di assistenza che ho mandato, ed una eventuale loro risposta | Sviluppo UI | Edoardo | 5 |
| | | | Design API | Edoardo | 5 |
| | | | Sviluppo API | Edoardo | 6 |
| | | | Testing | Carlo | 4 |
| | | | Documentazione | Gianlugi | 2 |
| | | | Deploy | Edoardo | 2 |
| | Accettazione ordine | Come utente farmacia voglio poter visualizzare un ordine in arrivo e decidere se accettarlo o rifiutarlo | Sviluppo UI | Carlo | 5 |
| | | | Design API | Carlo | 6 |
| | | | Sviluppo API | Carlo | 7 |
| | | | Testing | Edoardo | 4 |
| | | | Documentazione | Carlo | 2 |
| | | | Deploy | Gianluigi | 2 |
| | Visualizzazione consegne disponibili e accettazione | Come utente rider voglio poter visualizzare la lista di ordini in attesa di ritiro con relativi indirizzi su mappe in modo da poter accettare la consegna preferita e successivamente confermare la messa a termine dell'incarico | Sviluppo UI | Gianluigi | 6 |
| | | | Design API | Gianluigi | 7 |
| | | | Sviluppo API | Gianluigi | 8 |
| | | | Testing | Edoardo | 5 |
| | | | Documentazione | Carlo | 5 |
| | | | Deploy | Gianluigi | 2 |
| | Visualizzazione guadagni | Come utente rider voglio poter visualizzare le statistiche dei guadagni | Sviluppo UI | Gianluigi | 4 |
| | | | Design API | Gianluigi | 4 |
| | | | Sviluppo API | Gianluigi | 4 |
| | | | Testing | Edoardo | 3 |
| | | | Documentazione | Carlo | 3 |
| | | | Deploy | Gianluigi | 2 |
| | Privilegi utente admin | Come utente admin voglio poter accedere ad azioni privilegiate come la manutenzione del sistema, visualizzazione statistiche e modifica delle impostazioni globali | Sviluppo UI | Carlo | 5 |
| | | | Design API | Carlo | 4 |
| | | | Sviluppo API | Carlo | 5 |
| | | | Testing | Edoardo | 3 |
| | | | Documentazione | Gianluigi | 2 |
| | | | Deploy | Carlo | 1 |
| | Check out: invio ordine alla farmacia | Come utente, voglio inviare l'ordine alla farmacia per confermare il prezzo. | Sviluppo UI | Edoardo | 5 |
| | | | Design API | Edoardo | 5 |
| | | | Sviluppo API | Edoardo | 6 |

| | | | | | |
|--|---|---|----------------|-----------|---|
| | | | Testing | Carlo | 3 |
| | | | Documentazione | Gianluigi | 2 |
| | | | Deploy | Edoardo | 1 |
| | Controlli lista ordini candidati | Come farmacia, voglio vedere la lista degli ordini candidati per confermare disponibilità e prezzo. | Sviluppo UI | Carlo | 6 |
| | | | Design API | Carlo | 7 |
| | | | Sviluppo API | Carlo | 7 |
| | | | Testing | Gianluigi | 5 |
| | | | Documentazione | Carlo | 4 |
| | | | Deploy | Edoardo | 3 |
| | Conferma prezzo ordine candidato | Come farmacia, voglio confermare il prezzo aggiornato finale per un ordine candidato. | Sviluppo UI | Carlo | 5 |
| | | | Design API | Carlo | 5 |
| | | | Sviluppo API | Carlo | 7 |
| | | | Testing | Edoardo | 5 |
| | | | Documentazione | Carlo | 3 |
| | | | Deploy | Gianluigi | 2 |
| | Cambio stato ordine | Come farmacia, voglio aggiornare lo stato dell'ordine nelle varie fasi di selezione. | Sviluppo UI | Carlo | 5 |
| | | | Design API | Carlo | 5 |
| | | | Sviluppo API | Carlo | 6 |
| | | | Testing | Gianluigi | 4 |
| | | | Documentazione | Carlo | 3 |
| | | | Deploy | Edoardo | 2 |
| | Controllo ordini pagati in attesa del Rider | Come farmacia, voglio visualizzare gli ordini pagati in attesa del passaggio del Rider. | Sviluppo UI | Carlo | 5 |
| | | | Design API | Carlo | 6 |
| | | | Sviluppo API | Carlo | 6 |
| | | | Testing | Gianluigi | 4 |
| | | | Documentazione | Carlo | 2 |
| | | | Deploy | Edoardo | 1 |
| | Controllo secret code | Come farmacia, voglio verificare il codice segreto per confermare la consegna al Rider. | Sviluppo UI | Carlo | 4 |
| | | | Design API | Carlo | 5 |
| | | | Sviluppo API | Carlo | 6 |
| | | | Testing | Gianluigi | 4 |
| | | | Documentazione | Carlo | 2 |
| | | | Deploy | Edoardo | 1 |

| | | | | | |
|--|---|---|--|-----------|---|
| | Storico ordini farmacia | Come farmacia, voglio visualizzare lo storico degli ordini consegnati e in corso. | Sviluppo UI | Carlo | 6 |
| | | | Progettazione e implementazione DB (collezione ordini) | Edoardo | 5 |
| | | | Design API | Carlo | 6 |
| | | | Sviluppo API | Carlo | 8 |
| | | | Testing | Edoardo | 5 |
| | | | Documentazione | Carlo | 3 |
| | | | Deploy | Gianluigi | 2 |
| | Visualizzazione pool ordini Rider | Come Rider, voglio visualizzare la lista degli ordini che posso accettare. | Sviluppo UI | Gianluigi | 5 |
| | | | Design API | Gianluigi | 8 |
| | | | Sviluppo API | Gianluigi | 8 |
| | | | Testing | Edoardo | 5 |
| | | | Documentazione | Carlo | 3 |
| | | | Deploy | Gianluigi | 2 |
| | Accettazione ordine da parte del Rider | Come Rider, voglio accettare un ordine e visualizzare i dettagli per il ritiro. | Sviluppo UI | Gianluigi | 7 |
| | | | Design API | Gianluigi | 8 |
| | | | Sviluppo API | Gianluigi | 9 |
| | | | Testing | Edoardo | 4 |
| | | | Documentazione | Carlo | 3 |
| | | | Deploy | Gianluigi | 2 |
| | Cancellazione ordine da parte del Rider | Come Rider, voglio cancellare un ordine accettato per errore. | Sviluppo UI | Gianluigi | 3 |
| | | | Design API | Gianluigi | 4 |
| | | | Sviluppo API | Gianluigi | 5 |
| | | | Testing | Carlo | 3 |
| | | | Documentazione | Gianluigi | 2 |
| | | | Deploy | Edoardo | 1 |
| | Consegna avvenuta | Come Rider, voglio confermare l'avvenuta consegna dell'ordine. | Sviluppo UI | Gianluigi | 3 |
| | | | Design API | Gianluigi | 3 |
| | | | Sviluppo API | Gianluigi | 4 |
| | | | Testing | Carlo | 3 |
| | | | Documentazione | Gianluigi | 2 |
| | | | Deploy | Edoardo | 2 |
| | Middlewares sicurezza | Come utente, voglio essere reindirizzato alla pagina di | Sviluppo UI | Edoardo | 4 |

| | | | | | |
|--|---|---|----------------|-----------|-----|
| | | login se cerco di accedere a una pagina non autorizzata. | Design API | Edoardo | 6 |
| | | | Sviluppo API | Edoardo | 7 |
| | | | Testing | Gianluigi | 5 |
| | | | Documentazione | Carlo | 4 |
| | | | Deploy | Edoardo | 2 |
| | Accesso con Google - utente esistente | Come utente esistente, voglio autenticarmi tramite Google e accedere alla pagina corrispondente al mio ruolo. | Sviluppo UI | Carlo | 6 |
| | | | Design API | Carlo | 7 |
| | | | Sviluppo API | Carlo | 8 |
| | | | Testing | Gianluigi | 6 |
| | | | Documentazione | Carlo | 4 |
| | | | Deploy | Edoardo | 2 |
| | Autorizzazione - accesso a risorsa protetta | Come utente, voglio essere reindirizzato alla pagina di login se cerco di accedere a una risorsa protetta senza autenticazione. | Sviluppo UI | Edoardo | 4 |
| | | | Design API | Edoardo | 5 |
| | | | Sviluppo API | Edoardo | 6 |
| | | | Testing | Carlo | 4 |
| | | | Documentazione | Gianluigi | 3 |
| | | | Deploy | Edoardo | 2 |
| | Total | | | | 263 |
| | Ideal | | | | 263 |

Per maggiori informazioni sullo sprint backlog visualizzare il seguente [link](#)

3.2.2. Burndown Chart

Il burndown chart relativo a questo secondo sprint dimostra come l'impegno richiesto allo sviluppo del progetto, come nello sprint #1, si sia rivelato essere maggiore dell'andamento ideale soprattutto nelle prime fasi, per poi calare gradualmente verso i giorni finali,

concludendosi in pari con le aspettative.

Di seguito viene riportato il burndown chart relativo allo sprint #2

Burndown Chart



3.3. Test cases

Di seguito viene riportata una tabella, compilata con vari test-cases che abbiamo svolto, in modo da testare le API sviluppate durante questo secondo sprint. All'interno di questa tabella sono riportati i risultati che ogni operazione comporta, nel caso sia svolta correttamente, e gli errori che ci aspettiamo nel caso sia svolta in modo errato.

| Descrizione | Test Case Data | Precondizioni | Risultato atteso | Risultato reale | Note |
|--|--|--|--|--|------|
| Creazione di un account per un nuovo utente tramite email e password conforme. | Usare <i>api/v1/sign_up</i> e inserire una <i>email@domain.com</i> , una password: <i>Password1</i> , il Codice Fiscale, l'indirizzo di residenza e la data di nascita | Account <i>email@domain.com</i> non registrato | La mail non deve risultare esistente nel sistema | Account creato correttamente. Email non risultava esistente nel sistema. | |

| | | | | | |
|---|--|---|---|--|---|
| Creazione di un account per un nuovo utente con una email non valida. | Usare <i>api/v1/sign_up</i> e inserire una <i>email@domain.com</i> , una password: <i>Password1</i> , il Codice Fiscale, l'indirizzo di residenza e la data di nascita | Account <i>email@domain.com</i> già registrato | La registrazione viene negata a causa della mail già presente nel sistema | Registrazione negata. Messaggio di errore visualizzato: "Email già presente nel sistema". | L'utente viene avvisato del campo che genera il conflitto |
| Creazione di un account per un nuovo utente con un Codice Fiscale non valido. | Usare <i>api/v1/sign_up</i> e inserire una <i>email@domain.com</i> , una password: <i>Password1</i> , il Codice Fiscale, l'indirizzo di residenza e la data di nascita | Esiste già un account con questo Codice Fiscale | La registrazione viene negata a causa del codice fiscale già presente nel sistema | Registrazione negata. Messaggio di errore visualizzato: "Codice Fiscale già presente nel sistema". | L'utente viene avvisato del campo che genera il conflitto. Sia la mail che il CF sono campi chiave, quindi non duplicabili |
| Creazione di un account per un nuovo utente senza inserire l'email. | Usare <i>api/v1/sign_up</i> e inserire una password: <i>Password1</i> , il Codice Fiscale, l'indirizzo di residenza e la data di nascita | | Ogni campo del form risulta obbligatorio, quindi sia frontend che backend persistono nella obbligatorio del completamento dei campi | Segnalazione visiva all'utente. Richiesta non inviata a causa del campo email mancante. | Viene segnalato visivamente e all'utente e la richiesta non inviata |
| Registrazione di un nuovo utente con campo password vuoto | Usare <i>api/v1/sign_up</i> e inserire una <i>email@domain.com</i> , il Codice Fiscale, l'indirizzo di residenza e la data di nascita | | Ogni campo del form risulta obbligatorio, quindi sia frontend che backend persistono nella obbligatorio del completamento dei campi | Segnalazione visiva all'utente. Richiesta non inviata a causa del campo password vuoto. | |
| Login di un utente registrato con email e password corrette | Usare <i>api/v1/login</i> per effettuare il login nel sistema con <i>email@domain.com</i> , una | Account già registrato correttamente | Login avvenuto con successo e l'utente viene reindirizzato nella zona predisposta dalla sua tipologia | Login avvenuto con successo. Utente reindirizzato alla dashboard. | |

| | | | | | |
|---|---|--------------------------------------|---|---|------------------------------------|
| | password: <i>Password1</i> | | | | |
| Login di un utente registrato con email corretta ma con password errata | Usare <i>api/v1/login</i> per effettuare il login nel sistema con email <i>email@domain.com</i> , una password: <i>Wrong1</i> | Account già registrato correttamente | Login negato, l'utente avvisato con un banner rosso e indirizzato a eseguire nuovamente il login | Login negato. Banner rosso visualizzato. Utente indirizzato a eseguire nuovamente il login. | |
| Accesso di un utente già registrato utilizzando una mail sbagliata | Usare <i>/login</i> per effettuare il login nel sistema con email <i>Wrong@Wrong.com</i> , una password: <i>Password1</i> | Account già registrato correttamente | Login negato, l'utente avvisato con un banner rosso e indirizzato a eseguire nuovamente il login | Login negato. Banner rosso visualizzato. Utente indirizzato a eseguire nuovamente il login. | |
| Login di un utente registrato con email e password errate | Usare <i>api/v1/login</i> per effettuare il login nel sistema con email <i>Wrong@Wrong.com</i> , una password: <i>Wrong1</i> | Account già registrato correttamente | Login negato, l'utente avvisato con un banner rosso e indirizzato a eseguire nuovamente il login | Login negato. Banner rosso visualizzato. Utente indirizzato a eseguire nuovamente il login. | |
| Login di un utente con campi vuoti | Usare <i>api/v1/login</i> per effettuare il login nel sistema con email <i>""</i> , una password: <i>""</i> | | Ogni campo del form risulta obbligatorio, quindi sia frontend che backend persistono nella obbligatorietà del completamento dei campi | Segnalazione visiva all'utente. Richiesta non inviata a causa dei campi vuoti. | |
| Visualizzazione lista farmaci | <i>/api/v1/drugs</i> viene interpellato per scorrere i medicinali presenti nel sistema | | Caricamento dei primi 10 farmaci casuali scelti dal database da visualizzare all'utente | Primi 10 farmaci casuali caricati e visualizzati all'utente. | Possibile anche per utente anonimo |
| Ricerca del farmaco | Uso di <i>/api/v1/drugs/search?farmaco=\${encodeURIComponent(searchTerm)}</i> | | Ricerca tramite il campo <i>searchTerm</i> presente all'interno del form | Ricerca effettuata correttamente tramite il campo <i>searchTerm</i> . Risultati | Possibile anche per utente anonimo |

| | | | | | |
|---|--|--|---|--|---|
| | per la ricerca del farmaco | | | visualizzati. | |
| Aggiungi farmaco al carrello | Uso di <i>/api/v1/cart</i> per gestire il carrello, quale l'aggiunta di un farmaco | Utente loggato ricevente | Tramite il bottone <i>aggiungi al carrello</i> viene aggiunto all'array di prodotti del carrello associato all'utente | Farmaco aggiunto correttamente al carrello dell'utente. | <i>/api/cart</i> gestisce tutti gli aspetti per la visualizzazione e gestione del carrello elettronico |
| Rimuovere articoli dal carrello | Uso di <i>/api/v1/cart/remove</i> per rimuovere un articolo precedentemente inserito nel carrello | Utente loggato ricevente con dei prodotti nel carrello | Tramite un bottone rosso rimuovi, viene rimosso dall'array di prodotti del carrello, qualunque sia la quantità | Articolo rimosso correttamente dal carrello tramite il bottone rosso. | |
| Visualizzazione carrello | Uso di <i>/api/check-login</i> per verificare la presenza dell'utente loggato per la visualizzazione del carrello | Utente loggato ricevente | Accesso negato alla pagina in caso di login non effettuato correttamente | Accesso alla pagina del carrello negato in caso di login non effettuato. | Controllo aggiuntivo |
| Modifica quantità prodotti nel carrello | Uso di <i>api/cart/change</i> per modificare la quantità di un prodotto nel carrello | Utente loggato ricevente con dei prodotti nel carrello | Tramite dei tasti + e - è possibile variare la quantità di ciascun prodotto desiderato | Quantità dei prodotti modificata correttamente tramite i tasti + e -. | Se si diminuisce la quantità di un prodotto precedentemente pari a 1, questo viene rimosso dal carrello |
| Compilazione form assistenza | Uso di <i>api/v1/contact</i> per contattare l'assistenza fornendo in nome, mail, numero ordine e messaggio di riferimento del problema | | il campo numero d'ordine non è un campo obbligatorio. Il messaggio con i dati relativi all'utente viene correttamente registrato nel database | Messaggio con i dati utente registrato correttamente nel database. | |

| | | | | | |
|--|---|---|---|--|---|
| Aggiunta di un farmaco al carrello riservato dell'utente | Usare <i>api/v1/cart/add</i> per collegare il prodotto selezionato e inserirlo nel carrello corrispettivo dell'utente | Account loggato di tipo: <i>ricevente</i> | Articolo aggiunto correttamente al carrello. Visualizzazione pop-up di conferma | Articolo aggiunto correttamente al carrello. Visualizzazione pop-up di conferma | |
| Aggiunta di un farmaco al carrello riservato dell'utente, con un farmaco già presente nel carrello | Usare <i>api/v1/cart/add</i> per collegare il prodotto selezionato e inserirlo nel carrello corrispettivo dell'utente | Account loggato di tipo: <i>ricevente</i> | Articolo aggiunto correttamente al carrello. Visualizzazione pop-up di conferma. Se il prodotto è già presente la sua quantità viene incrementata automaticamente | Articolo aggiunto correttamente al carrello. Quantità del prodotto incrementata automaticamente. Pop-up di conferma visualizzato | Il prodotto sarà presente in quantità <i>n</i> volte |
| Modifica quantità carrello | Usare <i>api/v1/cart/change</i> per aggiornare la quantità | Account loggato di tipo: <i>ricevente</i> . Un prodotto presente nel carrello | Tramite il selettore "+" e "-" è possibile incrementare la quantità | Quantità del prodotto aggiornata correttamente tramite il selettore. Prodotto eliminato quando la quantità è stata portata a 0 | Se da 1 viene portata a 0, il prodotto viene eliminato |
| Rimozione articolo dal carrello | Usando <i>api/v1/cart/remove</i> | Account loggato di tipo: <i>ricevente</i> . Un prodotto presente nel carrello | Se la quantità del prodotto viene portata a 0, viene eliminato | Prodotto eliminato correttamente dal carrello quando la quantità è stata portata a 0 | |
| Ricerca di un farmaco nello shop | Uso <i>api/v1/drugs/search</i> per filtrare tra i risultati presenti nel database | Account loggato di tipo: <i>ricevente</i> . | Viene effettuata una ricerca cercando il nome nel <i>Farmaco</i> e <i>PrincipioAttivo</i> mostrando i primi 20 risultati | Ricerca effettuata correttamente. Visualizzati i primi 20 risultati in base al nome del farmaco e principio attivo | |
| Visualizzazione carrello per check out | Uso <i>api/v1/cart/details</i> e <i>api/v1/user/address</i> per realizzare la | Account loggato di tipo: <i>ricevente</i> . | Viene caricata la pagina riservata all'utente con i prodotti inseriti nel carrello nella fase di selezione dei | Pagina del carrello caricata correttamente con i prodotti inseriti. Possibilità di | C'è ancora la possibilità di poter eliminare i prodotti |

| | | | | | |
|---|---|---|--|---|---|
| | pagina del carrello | | farmaci | eliminare i prodotti visualizzata | prima del check out |
| Check out: invio ordine alla farmacia per conferma prezzo | Uso <i>api/v1/order/create</i> per finalizzare il processo di ordine, selezionando la farmacia desiderata e etichettando il tipo di ordine in: <i>inviato</i> | Account loggato di tipo: <i>ricevente</i> | Pop-up di conferma che l'ordine è stato inviato con successo | Pop-up di conferma visualizzato. Ordine inviato con successo e in attesa di conferma del prezzo dalla farmacia | Ora l'ordine deve essere confermato con il prezzo corretto dalla farmacia |
| Controlli lista ordini candidati | Uso <i>api/v1/ordini_f/candidati</i> | Account loggato di tipo: <i>farmacia</i> | Visualizzazione lista ordini riferiti alla farmacia loggata in attesa di conferma disponibilità prodotto/i e conferma prezzo finale escluse commissioni consegna | Lista ordini visualizzata correttamente. Ordini in attesa di conferma disponibilità prodotto/i e conferma prezzo | |
| Conferma prezzo ordine candidato | Uso <i>api/v1/ordini_f/id/aggiornaPrezzo</i> o per inserire e confermare il prezzo corretto per la farmacia | Account loggato di tipo: <i>farmacia</i> | Viene inserito il prezzo aggiornato finale, la farmacia lo conferma e un popup verde ne segnala la corretta trasmissione del prezzo. Una volta confermato verrà rispedito all'utente che potrà procedere al pagamento finale | Prezzo aggiornato e confermato correttamente. Pop-up verde visualizzato. Prezzo confermato rispedito all'utente per il pagamento finale | |
| Cambio stato ordine | Uso di <i>api/v1/ordini_f/id/cambiaStato</i> | Account loggato di tipo: <i>farmacia</i> | End point per aggiornare lo stato dell'ordine nelle varie fasi di selezione | Stato dell'ordine aggiornato correttamente nelle varie fasi di selezione | |

| | | | | | |
|---|--|--|--|---|--|
| Controllo ordini pagati in attesa del passaggio del Rider | Viene chiamato <i>api/v1/ordini_f/in corso</i> per vedere gli ordini pagati e presi in carico dal rider. | Account loggato di tipo: <i>farmacia</i> Ordine in stato: <i>Attesa</i> | Vengono visualizzati gli ordini per cui il rider è in fase di consegna, quindi è atteso a breve. | Ordini pagati visualizzati correttamente. Ordini in fase di consegna dal rider visualizzati | All'interno dello stesso pannello è presente la possibilità di effettuare la consegna per il singolo ordine mediante lo scambio e controllo del <i>secret code</i> |
| Controllo <i>secret code</i> | Uso <i>api/v1/ordini_f/orderId/verifyCode</i> | Account loggato di tipo: <i>farmacia</i> Ordine in stato: <i>Attesa</i> | Viene inserito il codice fornito dal rider all'operatore della farmacia che lo inserirà nell'apposito pannello di verifica | Codice fornito dal rider inserito correttamente. Prodotto segnato come consegnato al rider corretto dopo verifica del codice | Una volta confermato il prodotto risulterà consegnato al rider corretto |
| Storico ordini farmacia | <i>api/v1/ordini_f/storico</i> per visualizzare lo storico ordini riferito alla farmacia in essere | Account loggato di tipo: <i>farmacia</i> | Visualizzazione ordini in consegna o consegnati gestiti nel tempo dalla farmacia | Storico ordini visualizzato correttamente con ordini in consegna o consegnati gestiti nel tempo | |
| Visualizzazione pool ordini Rider | Visualizzazione <i>api/v1/orders</i> | Account loggato di tipo: <i>rider</i> | Visualizzazione lista ordini che il rider può accettare | Lista ordini che il rider può accettare visualizzata correttamente | |
| Accettazione ordine da parte del rider | Uso <i>api/v1/orders/:id/accept</i> | Account loggato di tipo: <i>rider</i> | Nel momento in cui l'ordine viene accettato, lo stato passa a: <i>attesa</i> | Ordine accettato correttamente. Stato passato a: <i>attesa</i> . Rider si reca in farmacia e preleva l'ordine grazie al <i>secret code</i> | Il rider si reca in farmacia, in cui grazie al <i>secret code</i> , univoco per ogni ordine, preleverà l'ordine |
| Cancellazione | <i>api/v1/orders/:id</i> | Account | Se il rider accetta | Ordine annullato | |

| | | | | | |
|--|---|--------------------------------------|---|---|--|
| e ordine da parte del rider | <i>/cancel</i> | loggato di tipo: <i>rider</i> | per errore un ordine può annullarne la presa in carico, facendo ritornare il medesimo ordine disponibile nella pool | correttamente dal rider. Ordine ritornato disponibile nella pool | |
| CONSEGNA AVVENUTA | | | | | |
| Middlewares sicurezza | Uso di <i>/api/v1/check-login</i> | Ogni account loggato | L'utente loggato o non, se per un motivo volontario o involontario finisce su una pagina non dedicata al suo ruolo, viene reindirizzato al login non avendo accesso alle risorse della pagina | L'utente loggato o non viene correttamente reindirizzato alla pagina di login senza accesso alle risorse della pagina non autorizzata | (es. utente rider finisce in pagine di farmacia) |
| Accesso con Google - utente esistente | Usare <i>/api/google</i> per effettuare l'accesso tramite Google | Utente esistente con account Google | L'utente viene autenticato e reindirizzato alla pagina corrispondente al suo ruolo | L'utente autenticato correttamente e reindirizzato alla pagina del suo ruolo | |
| Accesso con Google - nuovo utente | Usare <i>/api/google</i> per effettuare l'accesso tramite Google | Nuovo utente senza account esistente | L'utente viene registrato, autenticato e reindirizzato alla pagina corrispondente al suo ruolo | Nuovo utente registrato e autenticato correttamente. Reindirizzato alla pagina del suo ruolo | |
| Accesso con Google - fallimento autenticazione | Usare <i>/api/google</i> per effettuare l'accesso tramite Google e simulare un fallimento di autenticazione | Account Google non valido | L'utente viene reindirizzato alla pagina di errore <i>/loginFail.html</i> | Fallimento autenticazione simulato. Utente reindirizzato alla pagina <i>/loginFail.html</i> | |
| Logout - utente Google | Usare <i>/logout</i> per disconnettersi dal sistema | Utente loggato tramite Google | La sessione dell'utente viene terminata e l'utente viene reindirizzato alla homepage | Utente disconnesso correttamente. Reindirizzato alla homepage | |
| Autorizzazione - accesso a | Tentare di accedere a una | Utente non loggato | L'utente viene reindirizzato alla | Utente non loggato | |

| | | | | | |
|--|--|--|--|--|--|
| risorsa protetta | risorsa protetta senza essere autenticati | | pagina di login | reindirizzato alla pagina di login | |
| Successo nell'aggiornamento della password | Vecchia Password: <i>oldPassword123</i> Nuova Password: <i>newPassword123</i> | Utente autenticato Vecchia password corretta | Password aggiornata con successo Messaggio: "Password aggiornata con successo" Sezione nascosta | Password aggiornata con successo Messaggio: "Password aggiornata con successo" Sezione nascosta | Assicurarsi che le password rispettino i criteri di sicurezza |
| Password attuale non corretta | Vecchia Password: <i>wrongOldPassword</i> Nuova Password: <i>newPassword123</i> | Utente autenticato Vecchia password errata | Aggiornamento negato Messaggio: "Password attuale non corretta" | Aggiornamento negato Messaggio: "Password attuale non corretta" | Verificare che il messaggio sia chiaro e comprensibile per l'utente |
| Campi password vuoti | Vecchia Password: Nuova Password: <i>`newPassword123`</i> Vecchia Password: <i>`oldPassword123`</i> Nuova Password: | Utente autenticato Uno o entrambi i campi vuoti | Aggiornamento negato Messaggio: "Inserisci la vecchia e la nuova password" | Aggiornamento negato Messaggio: "Inserisci la vecchia e la nuova password" | Assicurarsi che il messaggio sia chiaro e comprensibile per l'utente |

3.4. Sprint review

Durante il nostro secondo sprint, abbiamo migliorato significativamente il nostro approccio allo sviluppo del servizio di consegna a domicilio dei farmaci. Nonostante l'inesperienza iniziale, siamo diventati molto più efficienti grazie ad un lavoro di coordinamento e informazione sui temi trattati. I daily scrum hanno migliorato la comunicazione e mantenuto il team allineato e concentrato sugli obiettivi.

Abbiamo fatto progressi notevoli nell'implementazione di diverse funzionalità chiave. Gli utenti possono ora aggiungere farmaci al carrello personale, modificarne le quantità e rimuovere articoli con gli appositi bottoni. La ricerca avanzata permette di trovare rapidamente farmaci specifici nel database.

Il processo di checkout è stato ottimizzato per consentire agli utenti di finalizzare gli ordini, con farmacie che possono confermare i prezzi e aggiornare lo stato degli ordini. I rider hanno accesso a una pool di ordini, con la possibilità di accettare, annullare e gestire le consegne in modo sicuro.

Abbiamo anche migliorato la sicurezza del sistema e integrato l'accesso tramite Google per semplificare l'autenticazione.

Per i prossimi sprint, intendiamo focalizzarci maggiormente sulle attività secondarie, migliorando tante piccole cose per una migliore esperienza utente. Continueremo a migliorare il nostro approccio agli sprint per affrontare le nuove sfide e rispondere meglio alle esigenze di un progetto in continua evoluzione. Questo ci permetterà di garantire un servizio di consegna dei farmaci ancora più efficiente e affidabile, rispondendo prontamente alle necessità dei nostri utenti.

3.5. Product backlog refinement

Durante la fase di Product Backlog Refinement, abbiamo cercato di identificare le funzionalità e le modifiche necessarie per migliorare l'esperienza utente e l'efficienza del sistema. Di seguito sono riportate alcune nuove funzionalità che abbiamo ritenuto essere utili per un futuro sprint:

1. Notifiche Push per gli Utenti

- **Descrizione:** Implementare un sistema di notifiche push che avvisi gli utenti (clienti, rider e farmacie) sulle modifiche dello stato degli ordini.
- **Impatto:** Miglioramento dell'esperienza utente e tempestività delle informazioni.
- **Demo:** L'utente riceve una notifica via email quando c'è un aggiornamento rilevante sull'ordine.

2. Sistema di pagamento

- **Descrizione:** Implementare un sistema di pagamento online tramite carta di credito o portafogli digitali (Apple Pay, PayPal...)
- **Impatto:** Miglioramento dell'esperienza utente e tempestività nell'ordinazione dei prodotti
- **Demo:** L'utente inserisce i dati della propria carta di credito e procede con il pagamento e la conferma definitiva dell'ordine

3. Sistema di Recensioni

- **Descrizione:** Consentire ai clienti di lasciare recensioni e valutazioni sui rider e sul servizio della farmacia. Tali recensioni possono essere viste dall'utente admin tra le statistiche per un'idea generale sulla qualità del servizio
- **Impatto:** Miglioramento della trasparenza e della qualità del servizio.
- **Demo:** Dopo la consegna, l'utente può lasciare una recensione e una valutazione direttamente nella sezione "I miei ordini"

4. Gestione delle Ricette Mediche

- **Descrizione:** Integrare un sistema per la gestione delle ricette mediche elettroniche, consentendo ai clienti di caricare le loro ricette e alle farmacie di verificarne la validità
- **Impatto:** Facilitazione del processo di ordine per farmaci su prescrizione.
- **Demo:** L'utente può caricare una ricetta medica che viene verificata dalla farmacia prima di confermare l'ordine.

3.6. Sprint retrospective

In questa riflessione, abbiamo esaminato i successi e le aree di miglioramento emerse durante il secondo sprint, con l'obiettivo di ottimizzare il carico di lavoro per il prossimo sprint. Ci concentreremo sul potenziare la produttività e la collaborazione tra i membri del team.

Durante questo sprint, abbiamo incontrato difficoltà significative nel reperire alcuni dati necessari, in particolare un sistema open data per la gestione e il controllo delle ricette mediche. Questo ostacolo ci ha impedito di realizzare tutte le implementazioni stimate all'inizio del progetto. Pertanto attualmente non è possibile gestire le ricette elettroniche all'interno del nostro servizio ma potrebbe essere parte di un implementazione futura.

Nonostante ciò, siamo riusciti a completare la filiera di gestione dell'ordine per utente ricevente, farmacia e rider, assicurandoci di coprire tutte le accortezze e i casi d'uso necessari.

Inoltre, abbiamo evitato di implementare un vero sistema di pagamento online in quanto complesso per questioni tecniche e di sicurezza, andando ad interfacciarsi con API di banche e richiedendo dati sensibili. L'integrazione di sistemi di pagamento richiede conformità con normative specifiche, protezione dei dati finanziari degli utenti e una gestione accurata delle transazioni, che comportano ulteriori risorse e competenze specifiche, attualmente al di fuori della nostra portata.

Ulteriori funzionalità che abbiamo lasciato da parte sono la gestione di un sistema di notifiche che possa avvisare i diversi utenti nelle fasi del processo, ad esempio quando un ordine è stato ritirato al rider o quando è stato accettato/rifiutato dalla farmacia, e un sistema di recensioni, in modo che i clienti possano valutare i rider. Queste funzionalità non sono state implementate in quanto abbiamo preferito concentrare le nostre risorse su altri aspetti, più rilevanti e funzionali per garantire un servizio quanto più completo possibile. Anche per questo caso non escludiamo di sviluppare queste funzionalità in una futura implementazione.

Anche durante questo secondo sprint, l'adozione delle pratiche Agile ha continuato a dimostrarsi vantaggiosa, permettendo una gestione flessibile ed efficiente dei compiti e facilitando il monitoraggio dell'avanzamento dello sprint.

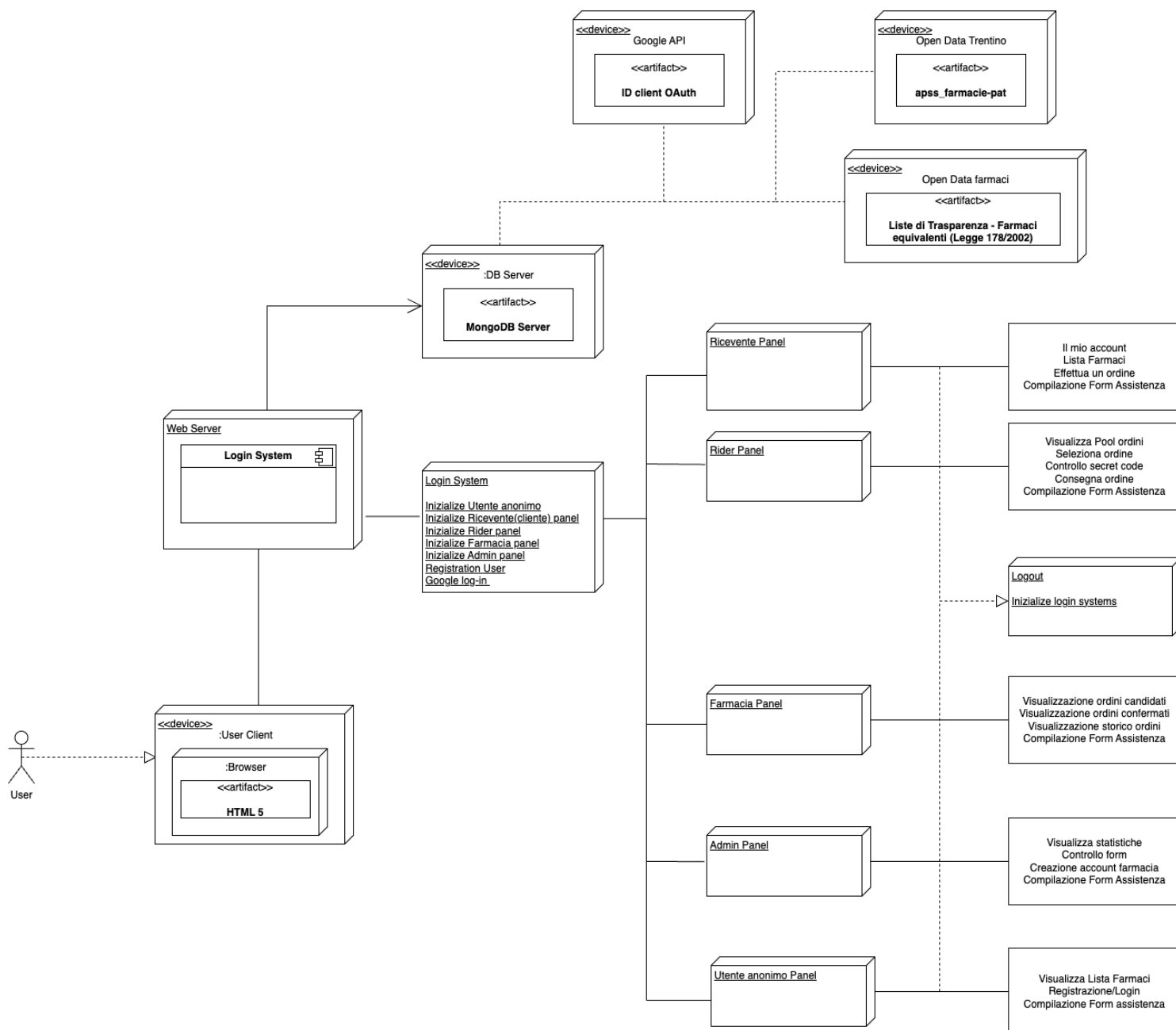
Un problema emerso durante questo sprint è stato il bilanciamento del carico di lavoro tra i membri del team, dovuto alle differenze nel background e nelle competenze individuali. Abbiamo affrontato questa sfida dedicando tempo per equilibrare il livello di competenze, lavorando individualmente per colmare le lacune e stabilire una base solida per i prossimi sprint.

Nonostante le sfide, il team ha mantenuto un bilancio pressoché in linea con le aspettative e i requisiti necessari, affrontando difficoltà e crescendo attraverso l'apprendimento autonomo e la sperimentazione pratica. Questo processo ha permesso a ciascuno di noi di esplorare nuovi argomenti e applicarli al progetto senza pressioni, rafforzando sia la coesione del gruppo sia le competenze individuali.

Guardando avanti, continueremo a migliorare la nostra pianificazione e gestione dei compiti per affrontare le sfide future con maggiore efficacia e adattabilità.

4. Sezione Finale

4.1. Diagramma del deploy finale con tutti i componenti (interni ed esterni)



4.2. Stack tecnologico utilizzato

Durante il nostro progetto, abbiamo impiegato un insieme di tecnologie che ci hanno permesso di sviluppare e implementare le funzionalità previste in modo efficiente e robusto. Ecco una panoramica del nostro stack tecnologico:

Frontend

- **Bootstrap:** Framework CSS per velocizzare il design e garantire una responsività cross-platform.

Backend

- **Node.js:** Ambiente di runtime per eseguire codice JavaScript lato server, scelto per la sua efficienza e scalabilità.
- **Express.js:** Framework web per Node.js, utilizzato per costruire l'API del nostro servizio e gestire le richieste HTTP. Utilizziamo *express-session* per la gestione delle sessioni.
- **MongoDB:** Database NoSQL utilizzato per la sua flessibilità nella gestione dei dati non strutturati e semi-strutturati.
- **Mongoose:** ODM (Object Data Modeling) per MongoDB, facilitando l'interazione tra l'applicazione Node.js e il database.
- **Morgan:** Middleware per il logging delle richieste HTTP, utile per monitorare e debuggare il traffico delle applicazioni.

Autenticazione e Sicurezza

- **OAuth2:** Protocollo utilizzato per la gestione del login tramite Google, garantendo un'autenticazione sicura e standardizzata.

Integrazione con Servizi Esterni

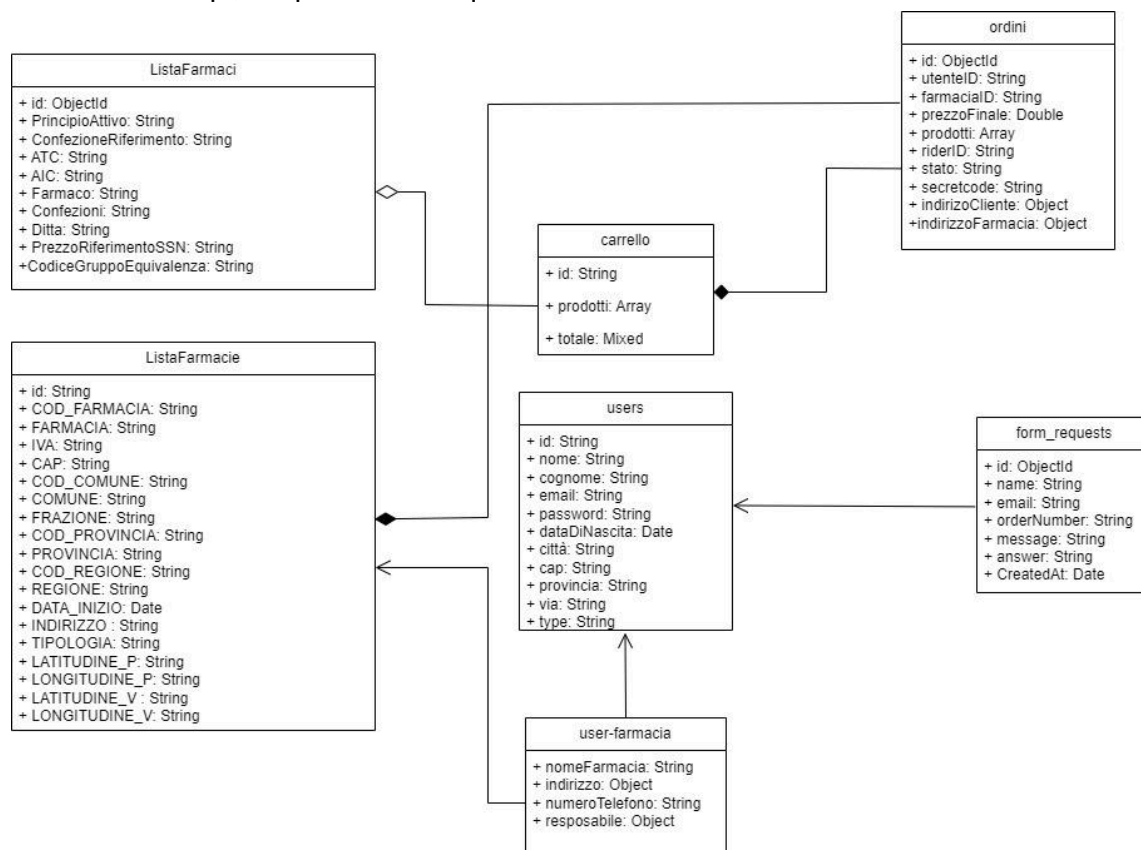
- **OpenStreetMap:** Utilizzato per l'integrazione delle mappe, permettendo la visualizzazione delle posizioni di consegna.
- **Google APIs:** Per il login e altri servizi di autenticazione.

Gestione del Progetto e Collaborazione

- **GitHub:** Piattaforma utilizzata per il versionamento del codice e la collaborazione tra i membri del team, con una strategia di branching robusta implementata durante il secondo sprint.
- **Visual Studio Code:** L'editor di codice utilizzato da tutti i membri del team per scrivere, modificare e debuggare il codice.

4.3. Aggiornamento Diagramma Entità Relazione

Di seguito riproponiamo un aggiornamento del diagramma entità relazione, riguardante esclusivamente lo sviluppo prodotto dal lavoro del team in questi mesi, ovvero la rappresentazione pratica presente nel nostro DataBase. Rispetto a quello proposto nel secondo deliverable sono presenti alcune modifiche che abbiamo dovuto intraprendere alla luce di riflessioni prodotte durante gli sprint e nel percorso di apprendimento della struttura delle api, in special modo quelle offerte dalla Provincia Autonoma.



4.4. Conclusioni

Negli ultimi tre mesi, il team ha affrontato la sfida di realizzare questo progetto, anche se nessuno dei membri aveva esperienza con queste tecnologie e metodi. È stato un viaggio che ci ha insegnato molto su come creare un team di sviluppo efficace e collaborare meglio.

Abbiamo lavorato insieme su implementazioni complesse, come i database dei farmaci e i servizi di autenticazione di Google, spesso in sede comune per ottenere risultati migliori. Grazie a una pianificazione flessibile e all'approccio Agile, siamo riusciti a completare la maggior parte delle task nonostante le difficoltà iniziali.

Completare la gestione dell'ordine per utenti, farmacie e rider, e sviluppare l'area admin sono stati traguardi importanti. Anche il sistema di login con Google ha migliorato la sicurezza e l'efficienza del servizio.

Questi tre mesi sono stati un'esperienza preziosa. Abbiamo imparato nuove competenze tecniche e siamo cresciuti come gruppo, migliorando la comunicazione e il lavoro di squadra. Siamo pronti a continuare su questa strada, affrontando nuove sfide con entusiasmo e spirito di collaborazione.